Logistic Regression (Week 3 lectures)

- classification
- y is discrete.

1. Classification

ex. email spam, online transactions, tumor.

· $y \in \{0, 1\}$ → binary class.
0 ↓ negative class
1 ↳ positive class.

· multi-class classification $y \in \{0, 1, \dots n\}$

· linear regression is not good for classification

1) although $y=0$ or 1, $h_\theta(x)$ can output out of range, i.e <0, or >1.

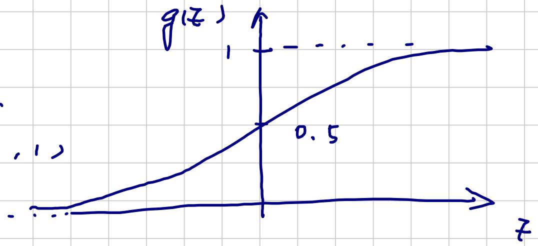
⇒ logistic regression : output always in-range. $0 \le h_\theta(x) \le 1$.
↓
classification.

2. Hypothesis Representation.

- want: $0 \le h_\theta(x) \le 1$. | linear regression. $\theta^T(x)$

⇒ $h_\theta(x) = g(\theta^T x)$, where $g(z) = \frac{1}{1+e^{-z}}$ (sigmoid function, logistic function)
↓ (z) real number.

$= \frac{1}{1+e^{-\theta^T x}}$.



hence satisfies $h_\theta \in (0, 1)$

pick value for $\underline{\theta}$.

Interpretation.

$h_\theta(x)$ = estimated probability of $y=1$ on input x.

ex. $\underline{x} = \begin{pmatrix} 1 \\ \text{tumor size} \end{pmatrix}$, $h_\theta(\underline{x}) = 0.7$ => 0.7% chance of being 1.

· $h_\theta(x) = P(y=1 \mid x ; \theta)$

↓ given x.

↳ parameterized by θ.

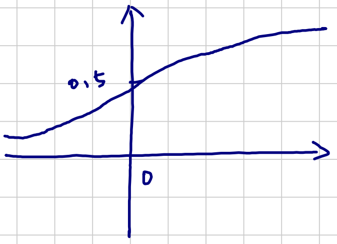
· $y = 0$ or 1. => $P(y=0 \mid x ; \theta) + P(y=1 \mid x ; \theta) = 1$.

3. Decision Boundary.

· $h_\theta(x) = g(\theta^T x)$; $g(z) = \frac{1}{1+e^{-z}}$

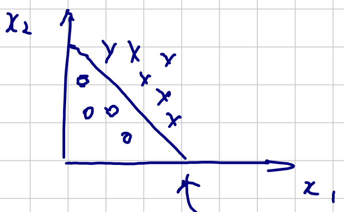
$= P(y=1 \mid x ; \theta)$

· Suppose predict $\begin{cases} \text{"}y=1\text{"} & \text{if } h_\theta(x) \geq 0.5, \quad \theta^T x \geq 0. \\ \text{"}y=0\text{"} & \text{if } h_\theta(x) < 0.5, \quad \theta^T x < 0 \end{cases}$



$g(z) \geq 0.5$ when $z \geq 0$,

$h_\theta(x) = g(\theta^T x) \geq 0.5 \iff \boxed{\theta^T x = 0}$



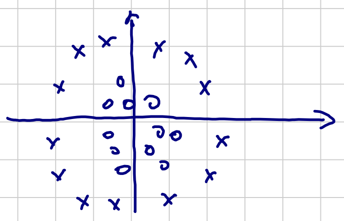
$h_\theta(x) = g(\theta^T \cdot x)$, say $\theta = \begin{pmatrix} -3 \\ 1 \\ 1 \end{pmatrix}$.

$\Rightarrow y=1$ when $\underbrace{(-3, 1, 1) \cdot \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix}}_{\text{equation of line}} \geq 0$.

ex. suppose $\underline{\theta} = \begin{pmatrix} 5 \\ -1 \\ 0 \end{pmatrix}$ s.t. $h_\theta(x) = g(\underbrace{5 - x_1}_{\text{line}})$

$5 - x_1 = 0 \Rightarrow x_1 = 5$

Non-linear decision boundaries.



$h_\theta(x) = (\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$

suppose $\underline{\theta} = \begin{pmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$

predict. "$y=1$" if $-1 + x_1^2 + x_2^2 \geq 0$.

$\Rightarrow$ equation of circle w/ radius 1

4. Cost function.

- define optimization objective. (cost function)

- training set. $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$

- m examples $x \in \begin{bmatrix} x_0 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1}$, $x_0 = 1$, $y \in \{0, 1\}$

- $h_\theta(x) = \dfrac{1}{1 + e^{-\theta^T x}}$. $\hookrightarrow$ how to choose $\underline{\theta}$?

- Recall:

  linear regression. $J(\theta) = \frac{1}{m} \cdot \sum_{i=1}^{m} \frac{1}{2} (h_\theta(x^{(i)}) - y^{(i)})^2$

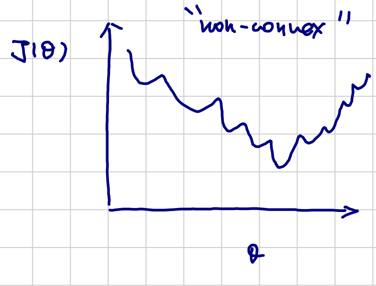
  $= \frac{1}{m} \sum_{i=1}^{m} \text{cost}(h_\theta(x^{(i)}), - y^{(i)})^2$

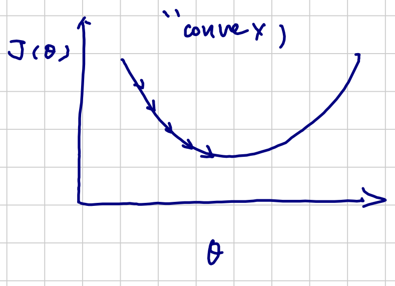
  $\Rightarrow \text{cost}(h_\theta(x^{(i)}), y^{(i)}) = \frac{1}{2} (h_\theta(x^{(i)}) - y^{(i)})^2$

  $\Rightarrow \text{cost}(h_\theta(x), y) = \frac{1}{2} (h_\theta(x), y)^2$

  $\hookrightarrow \dfrac{1}{1 + e^{-\theta^T x}}$. $\Rightarrow$ non-convex for "square cost function"

- Logistic regression.

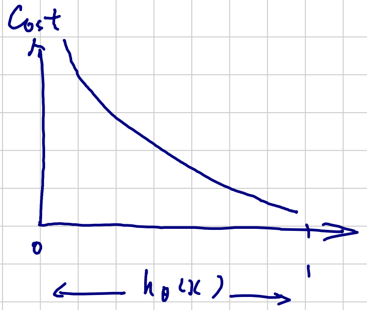




$\Rightarrow$ objective: create cost function s.t. $J(\theta)$ is convex.

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)), & y=1 \\ -\log(1-h_\theta(x)), & y=0. \end{cases}$$

if $y=1$ $\Rightarrow$



· cost $=0$ if $y=1$, $h=1$.

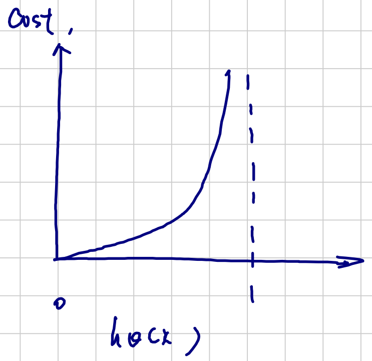
But as $h_\theta(x) \to 0$, cost $\to \infty$.

· captures intuition if $h_\theta(x)=0$. predict.

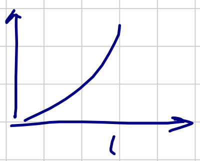
$P(y=1 \mid x;\theta)=0$ but $y=1$

↳ penalize learning algorithm w/ very large cost of ∞.

if $y=0$.



$-\log(1-z)$



5. Simplified cost function and gradient descent.

- $\text{Cost}(h_\theta(x), y) = -y \cdot \log(h_\theta(x)) - (1-y) \log(1-h_\theta(x))$

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \text{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$= -\frac{1}{m} \sum_{i=1}^{m} y \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))]$$

↳ maximum likelihood estimation.

To fit parameters θ, $\min_\theta J(\theta)$

To make prediction given new x, output $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$. $P(y=1 \mid x; \theta)$

- Gradient Descent.

$\min_\theta J$:

repeat {

$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$

}

$$\frac{\partial}{\partial \theta} J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (\underbrace{h_\theta(x^{(i)})}_{} - y^{(i)}) \cdot x^{(i)}$$

def'n has changed from linear regression!

- $\theta = \begin{pmatrix} \theta_0 \\ \vdots \\ \theta_n \end{pmatrix}$, $x^{(i)} = \begin{pmatrix} x_0^{(i)} \\ \vdots \\ x_j^{(i)} \\ \vdots \\ x_n^{(i)} \end{pmatrix}$ one sample

$j \to$ feature #.

- $\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)}$

$$\underset{(n+1)\times 1}{\underline{\theta}} = \underline{\theta} - \frac{\alpha}{m} \cdot \sum_{i=1}^{m} ((h_\theta(x^{(i)}) - y^{(i)}) \cdot x^{(i)})$$

Advanced optimization

. cost function. $J(\theta)$, want $\min_{\theta} J(\theta)$.

· gradient descent   Repeat $\{ \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \}$

Multiclass Classification.

ex. Email Foldering / Tagging $\rightarrow$ $y = \overset{\text{work}}{1}, \underset{\text{family}}{2}, \overset{\text{friends}}{3}, \underset{\text{hobby}}{4}$.

one-vs-all. (one-vs-rest)

- binary (this group or not).
- iterate through every group.

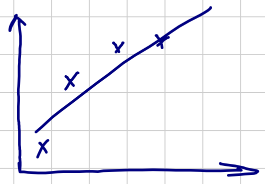
· fit 3 classifiers: $h_\theta^{(i)}(x) = P(y=i \mid x; \theta)$ $(i=1,2,3)$

↳ Train a logistic regression classifier. $h_\theta^{(i)}(x)$ for each class i to predict.

Lecture 7. Solving the Problem of overfitting.

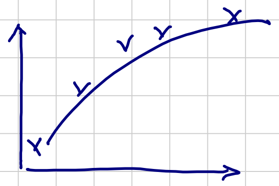
1. The problem of overfitting.

(1) Underfit, high bias



$\theta_0 + \theta_1 x$

(2) "Just right"



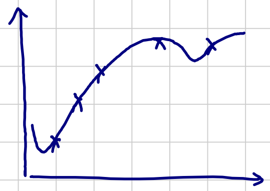
$\theta_0 + \theta_1 x + \theta_2 x^2$

(3) overfit, high variance

↓

too fitting, can fit any data

- too many features



$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Address overfitting

(1) Reduce number of features

- manually select which features to keep.
- model selection algorithm

(2) Regularization

- keep all the features, but reduce magnitude / values of parameters θ_j.

- Works well when we have a lot of features, each of which contributes a bit to predicting y.

2. Cost Function

· Intuition.

Suppose we penalize and make θ_3, θ_4 small. $\theta_3 \approx 0$, $\theta_4 \approx 0$

$$\hookrightarrow \min_{\theta} \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + 1000\,\theta_3^2 + 1000\,\theta_4^2$$

· small values for some θ_i ($i = 0, \dots n$.) $\longrightarrow$ simpler hypothesis

(small values for all θ_i) $\longrightarrow$ less prone to overfitting.

· Example - housing ① features: $x_1, x_2, \dots, x_{100}$

② Parameters: $\theta_0, \theta_1 \dots \theta_{100}$.

$\hookrightarrow$ we don't know what parameters to "shrink"

$\Rightarrow$ shrink all parameters.

$$J(\theta) = \frac{1}{2m} \Big[\underbrace{\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2}_{} + \lambda \sum_{j=1}^{n} \theta_j^2 \Big]$$

Ⓐ we want to train data fit

λ - regularization parameter. balances btwn objectives Ⓐ and Ⓑ.

θ_0 is excluded by convention

Ⓑ we want to keep θ small

· If λ is set too large, all $\theta_i \approx 0$, then $h_\theta(x) \approx \theta_0$.

$\hookrightarrow$ fit flat horizontal line

$\Rightarrow$ underfitting. (too high bias)

3. Regularized Linear Regression.

(1)

Gradient Descent

repeat {

$$\theta_0 := \theta_0 - \alpha \underbrace{\frac{1}{m} \sum_{i=1}^{m} [h_\theta(x^{(i)}) - y^{(i)}] x_0^{(i)}}_{\to \frac{\partial}{\partial \theta_0} J(\theta)}$$

$$\theta_j := \theta_j - \alpha \underbrace{\left\{ \frac{1}{m} \sum_{i=1}^{m} [h_\theta(x^{(i)}) - y^{(i)}] x_j^{(i)} + \frac{\lambda}{m} \theta_j \right\}}_{\frac{\partial}{\partial \theta_j} J(\theta) \text{, regularized}}$$

$j = 1, 2, 3 \ldots n$

}

$$\Rightarrow \quad \theta_j := \theta_j \underbrace{(1 - \alpha \frac{\lambda}{m})}_{1 - \alpha \frac{\lambda}{m} < 1 \text{.} \ \downarrow \ \text{small}} - \underbrace{\alpha \frac{1}{m} \sum_{i=1}^{m} [h_\theta(x^{(i)}) - y^{(i)}] x_j^{(i)}}_{\text{original gradient descent}}$$

① make θ_j a bit smaller

② perform regular update.

(2) Normal Equation.

Design Matrix. $\underline{X} = \begin{pmatrix} - (x^{(1)})^T - \\ \vdots \\ - (x^{(m)})^T - \end{pmatrix}_{m \times (n+1)}$ $\min_{\theta} J(\theta)$.

$y = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{pmatrix}$ $(m \times 1)$.

$\frac{\partial}{\partial \theta_j} J(\theta) \overset{set}{=} 0 \Rightarrow \theta = \left(X^T X + \lambda \underbrace{\begin{pmatrix} 0 & & & \\ & 1 & & \\ & & 1 & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix}}_{(n+1) \times (n+1)} \right)^{-1} X^T y$

e.g. $n=2$, $\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Non-invertibility.

Suppose $\underset{\#\text{examples}}{m} \leq \overset{\#\text{ features.}}{n}$ $\theta = (X^T X)^{-1} X^T y \rightarrow$ non-invertible / singular.

pinv. pseudoinverse
inv. inverse

If $\lambda > 0$, $\theta = \left(\underbrace{X^T X + \lambda \begin{pmatrix} 0 & & 0 \\ & 1 & \\ 0 & & 1 \end{pmatrix}}_{\text{invertible.}} \right)^{-1} X^T y$.

4. Regularized Logistic Regression.

$$J(\theta) = -\left(\frac{1}{m} \sum_{i=1}^{m} y^{(i)} \log \cdots \quad + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2 \right)$$

Repeat {

$$\theta_0 := \theta_0 - \frac{\alpha}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha \left(\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \right) + \frac{\lambda}{m} \theta_j$$

$j = 1 \dots n.$

}

$$\underline{h_\theta}(x) = \frac{1}{1+e^{\theta^T x}}.$$

$$\frac{\partial}{\partial \theta_j} \underbrace{J(\theta)}_{\text{regularized}}$$

Advanced Optimization.

$$\underline{\theta} = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{pmatrix} \begin{array}{l} - \text{theta}(1) \\ \vdots \\ \vdots \\ - \text{theta}(n+1) \end{array}$$

function [jVal, gradient]
= costfunction (theta).

↳ fminunc (@costfunction)

$$\frac{\partial}{\partial \theta_0} J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}.$$

$$\frac{\partial}{\partial \theta_j} J(\theta) = \left[\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)} - y^{(i)}) x_j^{(i)} \right] + \frac{\lambda}{m} \theta_j$$